

Name(English)	
Name(Arabic)	
Section	
Marks	Leave it blank

NOTE:

1. This is a multiple choice questions exam, you have to choose only one answer (**if not specified**) in each question (*choosing more than one answer makes the question ungraded*)
2. You have to answer all the questions (*there is no penalties upon the wrong answers*)
3. Make your choice by shading the answer circle and circle the answer itself
4. All codes you will see is upon this initialization
 - void init(void)
 - {
 - **glClearColor(1.0, 1.0, 1.0, 1.0);**
 - glViewport(0, 0, 320, 240);
 - **gluOrtho2D(-2 , 2, -2, 2);**
 - }
 - int main(int argc, char* argv[])
 - {
 - glutInit(&argc, argv);
 - glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
 - **glutInitWindowSize(600, 600);**
 - glutCreateWindow("Window");
 - init();
 - glutDisplayFunc(**Display**);
 - glutMainLoop();
 - return 0;
 - }

5. All Displays functions will begin with this code
 - `void Display(void)`
 - `{`
 - `glClear(GL_COLOR_BUFFER_BIT);`
 - `glColor3f(0.0, 0.0, 0.0);`
 - `glBegin(GL_LINES);`
 - `{`
 - `glVertex2f(-5, 0);`
 - `glVertex2f(5, 0);`
 - `glVertex2f(0, 5);`
 - `glVertex2f(0, -5);`
 - `}`
 - `glEnd();`
 6. All Displays functions will end with this code
 - `glFlush();`
 - `}`
 7. This test is not for testing your language skills, if you want to know the meaning of a word or a phrase, do not hesitate to raise your hand quietly and ask one of the TAs around you.
 8. Also, This exam is not for testing your noticing skills, every trick at this exam will be **bolded or underlined**.
 9. Finally, **Good Luck and Enjoy :)**
-

Questions:

1. **Computer graphics concerns with?**
 - Writing a program in a general purpose language to **produce** an image.
 - Writing a program in a general purpose language to **Manipulate** images and **Extract Information** from them.
 - **Draw Image** on the **wall Paper**.
2. **Image Processing concerns with?**
 - Writing a program in a general purpose language to **produce** an image.
 - Writing a program in a general purpose language to **Manipulate** images and **Extract Information** from them.
 - **Draw Image** on the **wall Paper**.
3. **Manipulating a picture on the photoshop is related to:**
 - Computer Graphics
 - **Image Processing**
 - None of them

4. Detecting your face when you upload a picture on facebook related to:
 - Computer Graphics
 - Image Processing
 - None of them
5. Creating character for a computer game related to:
 - Computer Graphics
 - Image Processing
 - None of them
6. Drawing a great picture on a white paper related to:
 - Computer Graphics
 - Image Processing
 - None of them
7. Which of those are application for computer graphics (may be there are more than one correct)
 - Design(CADs systems)
 - Take a selfie
 - Simulation and modeling (ex. Graphical flight simulator: real time graphics production,games, VR).
 - User interfaces
 - Detecting a criminal face **among many criminal pictures** by using information about the criminal.
 - Making a cupcake and delicious chocolate
8. why images are better displayed than text on CRT monitors ?
 - By nature, the illumination of the phosphor dots on the CRT monitors change **gradually** from pixels to the adjacent pixel.
 - the **abrupt** changes of the CRT makes the image clear.
 - the power of CRT illumination guns
9. why Texts are better displayed than images on LCD monitors ?
 - By nature, the illumination of the phosphor dots on the LCD monitors change **gradually** from pixels to the adjacent pixel.
 - the **abrupt** changes of the LCD makes the image clear.
 - the power of LCD illumination guns
10. what is the role of the graphics creator using Two dimensional drawing utilities and libraries ?
 - Forming image is done using the simple **two dimensional geometrical entities** (line, points, polygons) the libraries usually contains two types of functions and **the creator** should make the **projection** by him/herself.
 - Generating images is done through using a model for image generation that **simulates** optical imaging systems (cameras and human visual

system) and **Specifying what exists in the scene**. Also, the **computer** make the **projection**

- Use a paper and a pencil to draw a great picture and make a lot of money

11. what is the role of the graphics creator using Three dimensions utilities and libraries that utilizing scene/viewer/projection model?

- Forming image is done using the simple **two dimensional geometrical entities** (line, points, polygons) the libraries usually contains two types of functions and **the creator** should make the **projection** by him/herself.
- Generating images is done through using a model for image generation that **simulates** optical imaging systems (cameras and human visual system) and **Specifying what exists in the scene**. Also, the **computer** make the **projection**
- Use a paper and a pencil to draw a great picture and make a lot of money

12. The memory in a frame buffer must be fast enough to allow the display to be refreshed at a rate sufficiently high to avoid flicker. A typical workstation display can have a resolution of 1280 x 1024 pixels. If it is refreshed 72 times per second, That is, how much time approximately can we take to read one pixel from memory?

- **10.6 nanoseconds**
- 10.0 nanoseconds
- 54.3 nanoseconds
- 50.2 nanoseconds

13. in the Q12) What is this time for a 480 x 640 display that operates at 60 Hz but is interlaced?

- 10.6 nanoseconds
- 10.0 nanoseconds
- 54.3 nanoseconds
- 50.2 nanoseconds

هذا السؤال كان مجاب خطأ العام الماضي ولهذا جميع الخيارات خطأ
أما هذا العام تم تعديل الاجابة في السكشن وبالتالي الصواب هو
108.5 nanoseconds

14. Movies are generally produced on 35-mm film that has a resolution of approximately 2000 x 3000 pixels. What implication does this resolution have for producing animated images for a video show on a computer (480 x 640) as compared with film?

- It can take **20** times as much time to render each frame
- It can take **10** times as much time to render each frame
- It can take **50** times as much time to render each frame
- It can take **the same** as much time to render each frame

15. The role of the Primitive functions in OpenGL is:

- Draw **primitive** objects like points, lines, triangles and polygons
- specify how the objects are drawn. For examples, the functions that specify the current drawing **color**.

- specify how the objects are **viewed**. They specify the **location** of the viewer in the world, the view direction, and the type of **projection**.
- allow the user to **input** some **data** to the graphics program while being executed.

16. The role of the Input functions in OpenGL is:

- Draw **primitive** objects like points, lines, triangles and polygons
- specify how the objects are drawn. For examples, the functions that specify the current drawing **color**.
- specify how the objects are **viewed**. They specify the **location** of the viewer in the world, the view direction, and the type of **projection**.
- allow the user to **input** some **data** to the graphics program while being executed.

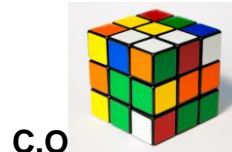
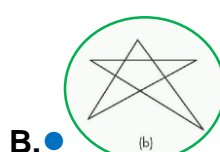
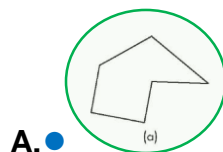
17. The role of the Viewing functions in OpenGL is:

- Draw **primitive** objects like points, lines, triangles and polygons
- specify how the objects are drawn. For examples, the functions that specify the current drawing **color**.
- specify how the objects are **viewed**. They specify the **location** of the viewer in the world, the view direction, and the type of **projection**.
- allow the user to **input** some **data** to the graphics program while being executed.

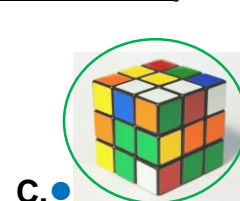
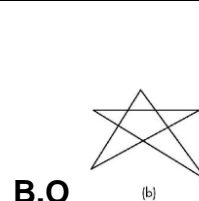
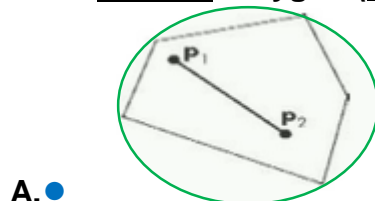
18. The role of the Attribute functions in OpenGL is:

- Draw **primitive** objects like points, lines, triangles and polygons
- specify **how** the objects are drawn. For examples, the functions that specify the current drawing **color**.
- specify how the objects are **viewed**. They specify the **location** of the viewer in the world, the view direction, and the type of **projection**.
- allow the user to **input** some **data** to the graphics program while being executed.

19. Choose the flat Polygon (*may be there are more than one answer*):

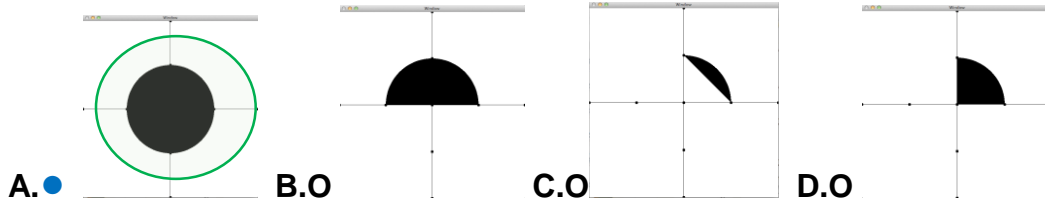


20. Choose the Convex Polygon (*may be there are more than one answer*):



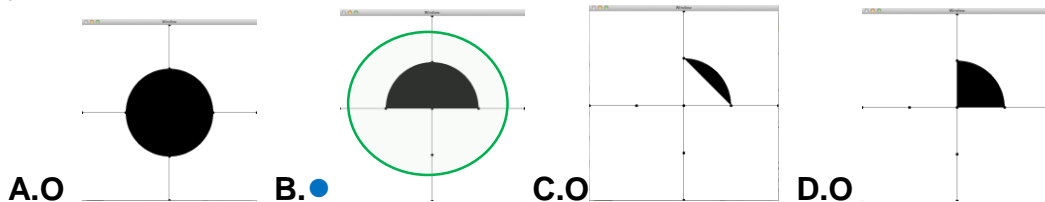
21. What this function will display ?

```
void func()
{
    glBegin(GL_POLYGON);
    for (double theta = 0.0 ; theta < 2 * PI ; theta += 0.001)
        glVertex2f(cos(theta), sin(theta));
    glEnd();
}
```



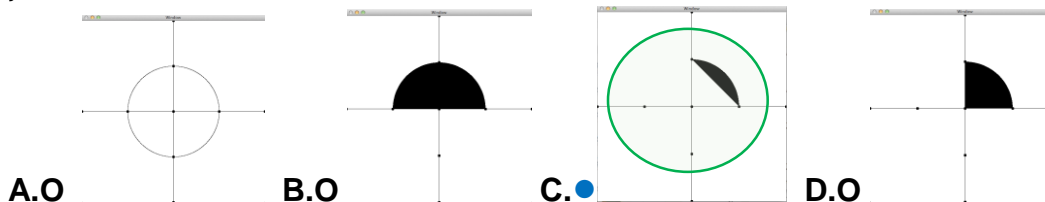
22. What this function will display ?

```
void func()
{
    glBegin(GL_POLYGON);
    for (double theta = 0.0 ; theta < PI ; theta += 0.001)
        glVertex2f(cos(theta), sin(theta));
    glEnd();
}
```



23. What this function will display ?

```
void func()
{
    glBegin(GL_POLYGON);
    for (double theta = 0.0 ; theta < 0.5 * PI ; theta += 0.001)
        glVertex2f(cos(theta), sin(theta));
    glEnd();
}
```



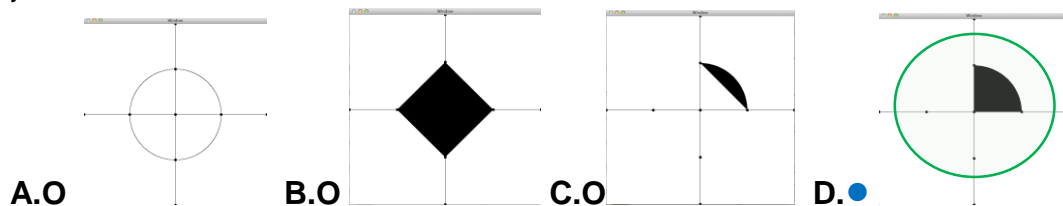
24. What this function will display ?

```
void func()
```

```

{
    glBegin(GL_POLYGON);
    for (double theta = 0.0 ; theta < 0.5 * PI ; theta += 0.001)
        glVertex2f(cos(theta), sin(theta));
    glVertex2f(0.0, 0.0);
    glEnd();
}

```

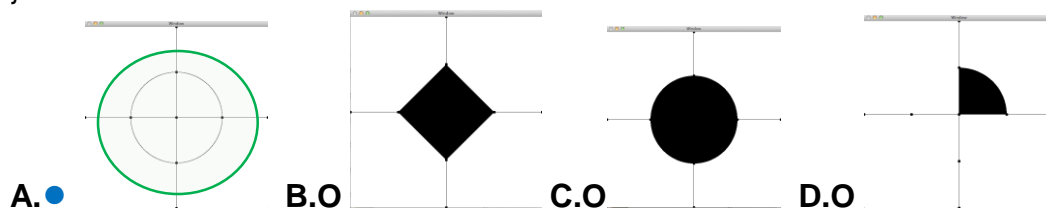


25. What this function will display ?

```

void func()
{
    glBegin(GL_POINTS);
    for (double theta = 0.0 ; theta < 2 * PI ; theta += 0.001)
        glVertex2f(cos(theta), sin(theta));
    glEnd();
}

```

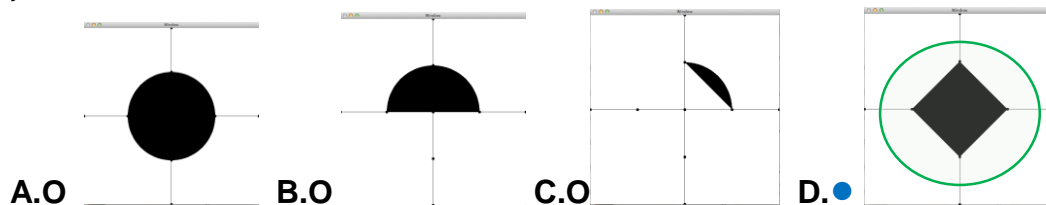


26. What this function will display ?

```

void func()
{
    glBegin(GL_POLYGON);
    for (double theta = 0.0 ; theta < 2 * PI ; theta += PI / 2)
        glVertex2f(cos(theta), sin(theta));
    glEnd();
}

```



27. Which function is responsible for creating the window ?

- ☐ glutInit(&argc, argv);

- glutInitWindowSize(200, 400);
- glutCreateWindow("Window Name");
- glutDisplayFunc(Display);

28. Which function is responsible for determining the window size ?

- glutInit(&argc, argv);
- glutInitWindowSize(200, 400);
- glutCreateWindow("Window Name");
- glutDisplayFunc(Display);

29. What this function will display ?

void func(GLfloat *a , GLfloat *b , GLfloat *c)

```
{
    glBegin(GL_POLYGON);
        glVertex3fv(a);
        glVertex3fv(b);
        glVertex3fv(c);
    glEnd();
}
```

- Triangle
- Circle
- Square
- Cube
- Rectangle

Good Luck and Best Wishes